Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4873574, 20 pages
https://doi.org/10.1155/2021/4873574

Research Article

CGPP-POI: A Recommendation Model Based on Privacy Protection

Gesu Li,[1] **Guisheng Yin**,[1] **Zuobin Xiong**,[2] **and Fukun Chen**[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Heilongjiang, China*
[2]*Department of Computer Science, Georgia State University, GA, USA*

Correspondence should be addressed to Gesu Li; lgs7788@hrbeu.edu.cn

Received 17 June 2021; Accepted 18 August 2021; Published 31 August 2021

Academic Editor: Yan Huo

At present, with the popularization of intelligent equipment. Almost every smart device has a GPS. Users can use it to obtain convenient services, and third parties can use the data to provide recommendations for users and promote relevant business development. However, due to the large number of location data, there are serious data sparsity problems in the data uploaded by users. At the same time, with great value comes great danger. Once the user's location information is obtained by the attacker, severe security issues will be caused. In recent years, a lot of researchers have studied the recommendation of point of interests (POIs) and the privacy protection of location. Yet, few of them have explored both together, which induces some drawbacks on the combination of them. This paper combines POI recommendation with a privacy protection mechanism. Besides providing user with POI recommendation service, it also protects the privacy of user's location. We proposed a POI recommendation model with privacy protection mechanism, termed POI recommendation model for community groups based on privacy protection (CGPP-POI). This model can ensure the recommendation accuracy and reduce the leakage of user location information via taking advantages of the characteristics of location. At the same time, it deals with the problem of poor recommendation performance caused by sparse data. In addition, through the expansion of location, random and other methods are used to protect the user's real check-in information. First, the data processed at the terminal satisfied local differential privacy. At the same time, we use the data to build a recommendation model. Then, we use a community of user in the model to improve the availability of these disturbed data, explore the relationship between users, and expand check-ins within the community. Finally, we provide the POI recommendations to users. Based on the traditional evaluation criteria, we adopted four metrics, i.e., accuracy, recall rate, coverage rate, and popularity in evaluation part, where intensive experiments conducted on real datasets Gowalla and Brightkite demonstrate that our approach outperforms the baseline methods significantly.

1. Introduction

With the advent of 5G techniques, people's lifestyles have been changed thoroughly. Smartphones have become a ubiquitous part in our daily life, e.g., using smartphone to seek information, shopping online, and performing navigation. Hou et al. [1] pointed out that with the help of smart devices, sharing information has become a normal part of people's life. Especially during the epidemic, intelligent equipment has a great impact on China, for example, scanning QR codes to report trips and taking online classes. In short, it has accelerated the rapid development of e-commerce industry and internet technology and driven the overall economic development. But the convenient life requires users to provide more information, and the most useful information is GPS location. Location information is a very special data, because it has bonded with people's routine, e.g., navigation, shopping, and social activity, and thus has strong private attributes. Almost all software and applications require users to provide GPS information if they want to get a better experience. Currently, in order to get recommendation service, users still need to upload their exact GPS data to third parties. According to this location, the third party can provide nearby researched results. The exploration of POI recommendations is still in early stages, while, with the rapid development, people need to get more fresh information in a shorter time. For

example, a person's daily pattern is two dots in a line, which means he/she is only active at home and work place. Suppose that one day he/she needs to go somewhere unfamiliar, searching on the internet for information is necessary. However, it is hard to figure out useful information among such a huge information flow. To solve this problem, recommendation system emerged, aiming at providing users with POI that they might be interested and saving searching time for them.

In recent years, recommendation research based on POI has focused on machine learning. Generally, the research on POI falls into the following categories: (1) tensor decomposition; (2) Markov chain model; and (3) neural network model. These POI recommendation algorithms have achieved good results in certain scenarios, but they lack generality. At the same time, they ignore the privacy issues caused by user trajectories. In the work of Xue et al. [2], the authors pointed that the friend relationship has an influence on users' future behavior, which shows that social information plays a role in the process of recommendation. Concurrently, the paper proposed that when excluding the influence of work location and family information, the influence of social relationship is more apparent. In addition, an attacker can infer a specific user by using friends and some background knowledge. Undoubtedly, social relationships indeed increase the risk of users' privacy leakage.

The development of GPS positioning and network technology not only brings convenience to users' lives but also increases the risk of users' location information leakage. Liang et al. [3] demonstrate that the prediction accuracy of tap position inference can be at least 90 percent by utilizing convolutional neural networks. As special information, location can be regarded as both public information and user's private information. For example, on a map, where every location is public, this information is available, and third parties can use this information and certain statistics to build user profiles. But there is no doubt that a specific user's trajectory is private. Once leaked, it will cause security and privacy risks to the user. If the leakage happens in the data collection stage, it will cause a huge disaster for both the company and the user, such as the AOL [4] and Netflix [5] data breaches, which cause immeasurable social and economic consequences.

For the preceding reasons, we should take into account the particularity of location information. We propose a POI recommendation model based on privacy protection, which explores the relationship between users and the range of check-in. For example, in Figure 1, there are three users, represented by red, yellow, and blue dots. Each user has a certain number of check-ins in a certain period of time. Intuitively, blue users and yellow users are more closely related because they are more likely to be in the same area. As we can see from the figure, each user's check-in is their exact coordinates. If the same check-in map is obtained by an attacker, the attacker can easily predict where a user will go in the future. But if we expand the scope of a user's check-in location, the amount of information it contains will be ambiguous. Take the large blue check-in in Figure 1 as an example. Within radius r_1, the check-in scope contains only one check-in. When it was extended to r_4, it had four check-ins. Therefore, if the r_4 range is uploaded for service, the probability of an attacker inferring a true check-in will be decreased. Therefore, the user's real location can be protected. Again, all the check-ins in Figure 1 and where they are located can be considered public information without identifying the user who visited them. This public information is useless to the attacker. But for users, these check-ins can provide them with more options for future utility. In addition, on the establishment of the recommendation model, we hope to enhance the relevance among users by expanding the scope of real check-in, as the blue and yellow users in Figure 1. However, when the scope is extended, as shown in the figure, many check-ins of yellow user and blue user are partitioned into the same range, which increases the connection between blue user and yellow user and adds stronger correlated options to the prediction. It also shows that in complex network, community is a suitable way to solve the relationship between users [6]. In this paper, we build a satisfactory mechanism to provide users with POI recommendations through the user check-in information, while preserving user privacy protection for location information. Experiments show that this model can provide users with good recommendation choices under the privacy protection mechanism. Meanwhile, this model is constructed in a hierarchical way which reduces the amount of computation and effectively reduces the time cost. The innovation points of this paper are as follows:

(i) Integrating community detection mechanisms with location network. The community detection model is improved according to the needs of this paper and is adapted to suit the location network. At the same time, it can solve the problem of data sparsity in location recommendation

(ii) Instead of the traditional knowledge graph built by all users, the knowledge graph is processed hierarchically. We explore the relationships between users, locations, and check-ins within the community

(iii) In the process of recommendation, privacy protection mechanism is considered. To deal with real location coordinates, we apply local differential privacy on it. At the same time, the privacy protection mechanism is also added at the end of the recommendation, protecting the user's real location and future travel safety.

The content of this paper is organized as follows: Section 2 is related work, discussing the current research status of POI from two aspects of recommendation model and privacy protection. Section 3 is the preliminary knowledge introduction of the model. The basic knowledge of community detection and LDP involved in the model is introduced. Section 4 is the problem definition of the model. Section 5 is the main part of the model, which is elaborated from two parts: privacy protection mechanism and recommendation. Section 6 is the experiment of the paper, including the introduction of data set, setting, evaluation standard, and the

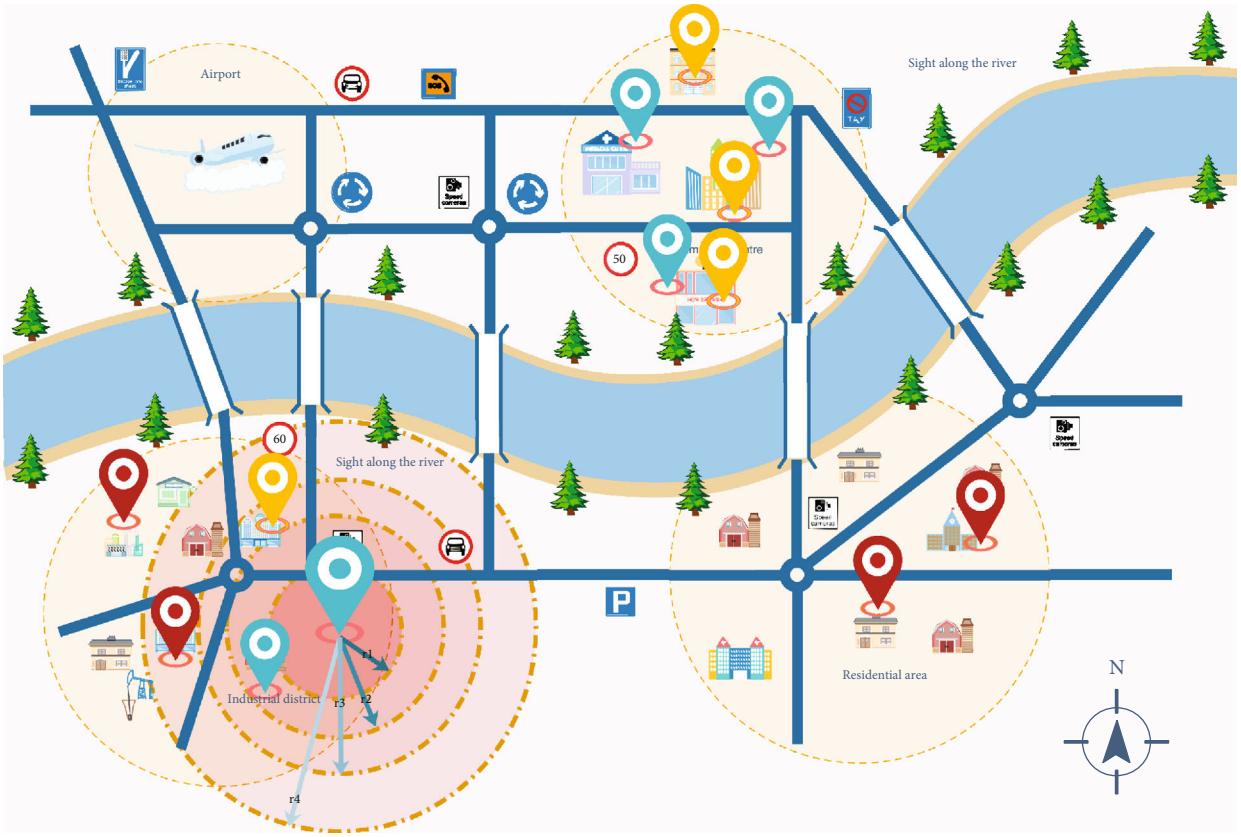


Figure 1: The example of users check-ins.

analysis of experimental results. Finally, the conclusion and future work of this paper is summarized in Section 7.

2. Related Work

In this section, we introduce the research on recommendation and privacy protection based on POI in recent years. We will discuss the current research status of these two aspects and point out the existing problems and the direction for improvement in this paper.

2.1. The Recommendation Model on POI. Recommendation based on POI has attracted much attention in recent years. Many scholars focus on solving the problem of data sparsity. In [7, 8], the authors mentioned that the density of check-in data is usually around 0.1%, while the data density of Netflix movie recommendations is around 1.2%. Data sparsity has a greater impact on POI recommendations. Zhao et al. [9] use context to predict the next POI through associative learning, in which authors construct an STGN to construct personalized sequences for users' long-term and short-term preferences. Li et al. [10] proposed a dynamic network recommendation algorithm based on HMM, which also includes privacy protection mechanism. Collaborative filtering, as a widely used recommendation algorithm, is also widely used in POI. Koren et al. [11–13] recommend algorithms based on matrix decomposition and its variants. Huang et al. [14] proposed a retrieval algorithm based on convolutional neural network, which is a recommendation technology for 3D and other Internet of Things devices, wherein Mnih and Salakhutdinov [13] proposed a PMF model containing Gaussian noise based on Bayesian framework. This model can maximize the posterior probability of potential features under the Gaussian hypothesis. Yin et al. [15] proposed an implicit probability generation model for the phenomenon of interest drift across geographic regions. This model can learn individual interests according to the check-in of individual interests in each region. It used social and spatial information to enhance regional dependence. At the same time, the author designed an effective pruning algorithm to overcome the dimension problem. Wang et al. [16] proposed a trust-based probabilistic recommendation model for social networks. In order to solve the problem of users' cold start, the authors consider their latent factor to find potentially similar users. The work of Li et al. [17] proposed a recommendation algorithm based on community division, which can also protect the privacy of users. At present, researchers have proposed a variety of solutions to the problem of data sparsity. With further understanding, most methods improve accuracy by adding features and increasing the complexity of the model. The result is an increased cost of time, which also requires users to upload more personal information when using third-party software to access the corresponding services. This is easy to cause the leakage of privacy.

2.2. The Privacy Protection on POI. Location-based services typically allow users to upload their current location first.

The third party obtains the information and provides recommendation service for the user according to the surrounding situation. Usually, the information uploaded by the user is exact location coordinates. In this paper, we denote this exact coordinate as L. But when a user provides location to a third party, there is no way to determine whether the third party can be trusted. At the same time, there is a risk of leakage during the process of information collection. Therefore, researchers have proposed some solutions to protect users' location privacy. K-anonymity is the most widely used privacy protection method. Kido et al. and Shankar et al. [18, 19] proposed a method to protect the user's identity information, so that the attacker cannot deduce which specific user is querying. Xue et al. and Bamba et al. [2, 20] focus on protecting the user's real location, where authors hide the user's real location by using virtual location. The preceding privacy protection methods can be boiled down to hiding. By hiding real data in other data, it prevents the attacker from obtaining the user's real location. There are other privacy protective mechanisms. Cover [21] introduced a stratagem-based protection mechanism. Papadopoulos et al. and Ghinita et al. [22, 23] proposed methods based on private information retrieval. Cai et al. [24] proposed a data sanitization method that collectively manipulates user profile and friendship relations. Besides sanitizing friendship relations, the proposed method can take advantages of various data-manipulating methods. Memon et al. [25] proposed a multimixed region privacy protection method which is dedicated to the mapping service of vehicles on the road network. Hashem and Kulik [26] proposed the use of mobile devices to form personal self-organizing networks. The authors proposed a decentralized method to protect the privacy of visitors' location. Chen et al. [27] proposed a method to protect user location privacy based on unobservability, in which the author designed, implemented, and evaluated a protection system named LISA, which is a location information scrambler. The system can adjust the location noise, protect the user's location privacy, and save the resources (especially the power) on mobile devices. Although these privacy protection mechanisms can protect the user's location security to a certain extent, they are lying under the assumption that the background knowledge of the attacker is limited, thus with some limitations. Some scholars have proposed spatial transformation, namely encryption, to protect the user's location information, yet these approaches are difficult to implement on mobile devices because of the huge computation cost even though the bandwidth has been increased with the help of 5G technology.

Moreover, more researches [28–32] have investigated the privacy protection problem in the background of 5G, IoT, and Big Data. As a representative of them, differential privacy [33], a privacy concept in the field of statistical databases, has been widely adopted. Its goal is to publish statistics about the database while protecting users' personal data. Differential privacy requires that modifying the data of a single user has negligible impact on the query results. Zhao et al. [34] proposed a top-down partitioning algorithm based on local differential privacy (LDP). It can generate undifferentiated location record data. At the same time, a new adaptive user assignment scheme and a series of optimization techniques are used to improve the accuracy of published data. It can be seen from the above studies that some researchers usually adopt mechanisms such as anonymity and disturbance to protect data privacy, and others also adopt encryption or differential privacy. This paper integrates multiple privacy protection mechanisms. We build protection mechanisms for data in the aspect of query, publication, and prediction, which can fully protect the location privacy of users.

3. Preliminaries

In this section, we introduce two main technologies covered in this paper, i.e., community detection and local differential privacy.

3.1. Community Detection. Community detection is typically applied to complex networks such as social networks [35], where each user is represented as a node. It focuses on the relationship between users. Tight-knit users are often divided into a community [36]. At present, communities are divided into overlapping communities and nonoverlapping communities. Overlapping community is more consistent with the division of communities in real life. Therefore, it is widely used. This paper makes an improvement on the traditional community detection by making it blend with the location characteristics. We choose the fast unfolding [37] algorithm as the basic algorithm of community detection and make improvement upon it. The algorithm has the advantages of fast computation and suitable for overlapping community division.

Fast unfolding algorithm is an iterative algorithm, which is mainly divided into two stages. The first stage is the module optimization stage. It mainly divides each node into the community where the adjacent nodes are located, thus increasing the value of modularity. The second stage is the community aggregation stage. It is to aggregate the communities divided in the first step into a single point. In other words, aggregation stage restructures the network. The algorithm is performed by repeating the above two steps until the network structure is stable. The formula involved is as follows:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{i,j} - \frac{k_i k_j}{2m}\right]\sigma\left(c_i, c_j\right), \tag{1}$$

where $m = 1/2\sum_{i,j}A_{i,j}$ is all the weights in the network, $A_{i,j}$ is the weight between u_i and u_j, and $k_i = \sum_i A_{i,j}$ is the weight of the edge linked to the vertex. c_i is the community to which the vertex is assigned. $\sigma(c_i, c_j)$ is used to determine whether u_i and u_j are divided into the same community. If it is divided into the same community, it returns 1. Otherwise, it returns 0. Whether it is divided into a community depends on whether ΔQ is positive or not. In other words, if a node is divided into the existing community, the change of modularity is positive.

3.2. Local Differential Privacy. Dwork et al. [38] introduced the concept of differential privacy in 2006. The initial work is concerned with the processing of central difference privacy [33, 38, 39]. Differential privacy is a privacy preserving statistical query method based on statistics. The purpose is that even if all record in the database is changed, the statistical result will not change too much. One problem with centralized differential privacy, however, is that the third party must be trusted. In fact, the third party is not necessarily trustworthy in reality. To solve this problem, local difference privacy (LDP) is proposed. In LDP, the data is processed by the user locally and then uploaded to the central server; therefore, it provides better privacy protection. The definition of LDP is as follows.

A randomized algorithm f is $\epsilon - \mathrm{LDP}$ if for any two check-in records t and t', $t, t' \in Dom(f)$, and for any possible output $t* \in Ran(f)$, the following equation is satisfied:

$$\Pr\left[f(t) = t^*\right] \leq e^{\epsilon} \times \Pr\left[f\left(t'\right) = t^*\right]. \tag{2}$$

The perturbation mechanism commonly used for LDP is random response technique. In LDP, each user perturbs their data and uploads it to the central processor. As a result, the data of any two users cannot be obtained from each other, so the concept of global sensitivity does not exist.

4. Problem Definition

The purpose of this paper is to predict POI for users in a recommendation model. Particularly, we use user location to build a recommendation mechanism. In addition to that, this model also needs to protect users' location privacy and to prevent an attacker from using all of the user's data to infer the real location. To facilitate better understanding of the model for readers, we define the concepts and formulate the problems involved in the model as follows.

Definition 1. Location network G. The location network graph based on user check-in is defined as G. $G = (V, E)$ is made up of nodes and edges, where V and E represent nodes and edges, respectively, wherein G is formed by n subgraphs. The n subgraphs are obtained by community detection. That is, each community forms a subgraph. $G = \{g_1, g_2, .., g_n\}, n \in |C|$. The node in the graph is user U.

$U = \{u_1, u_2, \cdots, u_i\}, i \in |U|$. Each user has its own check-in sequence. $u_i = L_{u_i} = \{l_1, l_2, \cdots, l_j\}, j = |L_{u_i}|, L_{u_i} \subseteq L$, where L is the check-in set of all user. However, in order to ensure the privacy of user location data, we mix up the order of check-ins before uploading. $L = (LOC, LOT)$, where LOC is the set of the check-in label in L, and LOT is the set of precise coordinates corresponding to the check-in label. The following formula is a summary of the relationships in the graph.

$$\begin{cases} G = (U, E) \\ G = \{g_1, g_2, \cdots, g_n\}, n = |C| \\ U = \{u_1, u_2, \cdots, u_i\}, i = |U| \\ u_i = \{l_1, l_2, \cdots, l_j\}, j = |L| \\ L = (LOC, LOT) \\ loc_m = (lat_m, lon_m). \end{cases} \tag{3}$$

Definition 2. $L, \widehat{L}, \tilde{L}$. In order to ensure the privacy of user location information, we mix up the order of check-ins before uploading. Here, we focus on the following three check-in definitions. (1) L is the real check-in set of users; (2) $\widehat{L}$ is the disturbed check-in set uploaded; and (3) $\tilde{L}$ is the expanded check-in set of true check-in. The relationship between the three check-in sets is as follows.

$$\begin{cases} L = \{l_1, l_2, \cdots, l_j\} = \widehat{L} = \left\{\widehat{l}_1, \widehat{l}_2, \cdots, \widehat{l}_j\right\}, l_j \neq \widehat{l}_j, j = |L| \\ R = \{r_1, r_2, \cdots, r_m\}, m = |R| \\ \tilde{l}_q = l_j \times r_m = \{l_1, l_2, \cdots, l_k\}, l_j \in L, r_m \in R, \tilde{l}_q \subseteq L \\ P\left(\widehat{l}_j = l_k\right) = \dfrac{1}{\left|\tilde{l}_q\right|}, l_k \in \tilde{l}_q \\ \tilde{L} = \left\{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_q\right\}, q = \left|\tilde{L}\right| \leq L, \end{cases} \tag{4}$$

wherein the extended check-in set $L\sim$ is all the check-ins within this range obtained by taking the real check-ins l_j as the center and r_m as the radius. The perturbation check-in $l\hat{}_j$ is the replacement check-in randomly selected within the perturbation range of the real check-in. So, even though the set of true check-ins and the set of perturbed check-ins have the same content, each perturbed check-ins are a random substitution of its corresponding true check-ins. Take u_i as an example. L_{ui} and $L\hat{}_{ui}$ are both subsets of the check-in set, but these two sets are not the same. The elements in the perturbation set $l_{u\hat{}i,j}$ are randomly selected and replaced from all the check-ins within the radius of r_m centering on the real check-in $l_{ui,j}$. $l_{u\hat{}i,j}$ is randomly selected from the extended set, used to replace the original check-in l_j. The other real check-ins of u_i are replaced by the above method. Finally, the perturbation set $L\hat{}_{ui}$ of u_i is obtained.

$$\begin{cases} \widehat{l}_{u_i,j} \in \widehat{L}_{u_i}, \widehat{l}_{u_i,j} \in \widehat{l}_{u_i,j} \\ L_{u_i} \neq \widehat{L}_{u_i} \\ L_{u_i}, \widehat{L}_{u_i} \subseteq L \\ \tilde{l}_{u_i,j} = l_{u_i,j} \times r_m = \{l_1, l_2, \cdots, l_k\}, \tilde{l}_{u_i,j} \subseteq \tilde{L}, l_{u_i,j} \in L_{u_i} \\ P\left(\widehat{l}_{u_i,j} = l_k\right) = \dfrac{1}{\left|\tilde{l}_{u_i,j}\right|}, l_k \in \tilde{l}_{u_i,j}. \end{cases} \tag{5}$$

Definition 3. Similarity. Community detection is based on close relationships between users. We use the similarity between users as the weight, which is used to indicate the closeness of the relationship between users. The higher similarity implies the closer relationship, and the more likely it is to be divided into the same community. Specially, the Jaccard similarity index is used to measure the similarity between two users.

$$Sim(i,j) = \frac{\tilde{L}_{u_i} \cap \tilde{L}_{u_j}}{\tilde{L}_{u_i} \cup \tilde{L}_{u_i}}. \tag{6}$$

In Equation (6), $Sim(i,j)$ represents the similarity between u_i and u_j. $\tilde{L}_{u_i}, \tilde{L}_{u_j}$ are the check-ins of extended scopes u_i and u_j, respectively. The numerator is the check-in that both u_i and u_j have visited, and the denominator is the check-in set of two users.

Definition 4. Community. The first step of community detection is to establish an adjacency matrix between users. Different from existing works, this paper takes users and their check-in as the research object. There is no directionality between users. At the same time, the similarity of check-in between users replaces the weight of edge in traditional social network. That is, the higher the similarity between u_i and u_j, the easier it is to be divided into the same community. The node of the community is the user. C is the set of all communities. $C = \{c_1, c_2, \cdots, c_n\}, n \in |C|$. $c_n = \{u_i, \cdots, u_j\}$. According to the definition of community detection in this paper, the formula of modularity is improved, and the improved formula of modularity is as follows:

$$Q = \frac{1}{2m}\sum_{i,j}\left[Sim(i,j) - \frac{\sum_j Sim(i,*)\sum_i Sim(j,*)}{2m}\right]\sigma(c_i, c_j). \tag{7}$$

Definition 5. Incidence matrix. M is the relationship matrix between the user and the extended check-in $\tilde{L}$. The weight in M is the number of paths that the user reaches $\tilde{L}$ after passing k hops. Since each community is a subgraph, the relationship can be formulated as follows.

$$\begin{cases} m_n \subset M, n \in |C| \\ M = U \times \tilde{L}. \end{cases} \tag{8}$$

Definition 6. Privacy. For any user, there is a privacy algorithm f. Its domain is $Dom(f)$, and range is $Ran(f)$. If the algorithm f gets the same output t^* on any two records t and t', it is said that function f satisfies ϵ – LDP.

$$\Pr[f(t) = t^*] \le e^{\epsilon} \times \Pr\left[f\left(t'\right) = t^*\right]. \tag{9}$$

Then, the relationship amid perturbation check-in, radius, and target check-in is given as follows.

$$P\left(\hat{l}_j = l_k\right) = \frac{1}{\left|\tilde{l}_q\right|}, l_k \in \tilde{l}_q,$$
$$\tilde{l}_q = l_j \times r_m = \{l_1, l_2, \cdots, l_k\}.k = \left|\tilde{l}_q\right|, \tag{10}$$

wherein $\tilde{l}_q$ is the set of check-ins that l_j gets according to r_m. The intensity of privacy protection is determined by r_m and the number of check-ins contained within its scope, where the range of r_m is set by the user. In other words, the bigger range, the more check-ins in $\tilde{l}_q$, and the stronger the perturbation. $\hat{l}_j$ is a random selection of perturbation data from $\tilde{l}_q$ to replace l_j.

Since it is a random replacement of the original check-in, the randomness satisfies the random disturbance mechanism of LDP. This approach takes advantage of the scalability of the location. The statistical results within a certain range are hardly affected by the expansion of the range. That is, data characteristics are preserved. In other words, the data has good utility. According to this algorithm, $\epsilon = \ln(1/(|\tilde{l}_q| - 1))$ can be obtained. The data uploaded by each user is $\hat{L}_{u_i}$, which is the perturbed check-in. In the third-party statistical process, it satisfies the LDP, and the statistical results obtained after perturbation are basically the same as the real ones.

The specific derivation formula is given by the following example.

Suppose there are n users using the software, where the true percentage of users check-in at A is π. The third party hopes to get the true number of users who check in at A through calculating the data uploaded by users for answers. If the probability of A being checked in the true answer of the data uploaded by u_i is $1/k$, the probability of getting the other answer is $1/(k-1)$. According to the answers of n users, the number of people who have checked in A can be derived. If the statistics show that the number of people who have been to A is n_1, the number of people who get the other result is $(n - n_1)$. According to this statistic, the likelihood function LH is constructed.

$$LH = \left[\pi\frac{1}{k} + (1-\pi)\left(1 - \frac{1}{k}\right)\right]^{n_1}\left[(1-\pi)\frac{1}{k} + \pi\left(1 - \frac{1}{k}\right)\right]^{n-n_1}, \tag{11}$$

where the maximum likelihood estimate of π is obtained after the logarithmic derivative:

$$\hat{\pi} = -\frac{kn_1 + n - nk}{nk - 2n}. \tag{12}$$

By further solving the mathematical expectation of π, we can verify that the above estimation is an unbiased estimation of π. Therefore, the number of people who have been to A is calculated as follows:

$$N = \widehat{\pi} \times n = \frac{1-k}{2-k} n + \frac{k n_1}{2-k}. \quad (13)$$

Accordingly, based on the total number of people, we can get the number of people who went to A through perturbation probability. If we get that the number of people who have been to A is n_1, and the perturbation probability $1/k$, we can get the real number of users who have been to A. By definition, the privacy budget of this inference result is calculated by Equation (14):

$$\epsilon = \ln \frac{1}{k-1}, \quad (14)$$

where k is the number of elements in $\tilde{l}_j$, $k \in (1,+\infty)$.

5. CGPP-POI

As a special information, location data is of great significance to users. Using location properly can provide great convenience for users' daily life and can also bring more benefits to the third party. But location is also important privacy information because it records the user's trajectory. Once the information is obtained by the attacker, the user's private property and even personal safety will be threatened. Considering the particularity of location information, when we build the recommendation model, we should add privacy protection mechanism to the user location. In this work, the privacy of users is protected from three aspects: publication, inquiry, and recommendation. We propose a recommendation model based on privacy protection, namely the CGPP-POI model. This model builds the privacy protection mechanism based on LDP, which can effectively prevent the leakage of real location. At the same time, by extending the original location range, the coincidence degree of activity range among users is improved. Moreover, the user community is divided based on this, leveraging communities to extend user data to address sparse data and provide users with POI recommendation service. Figure 2 shows the overall framework of CGPP-POI, where the operation of LDP is in the blue dotted box. This part processes the original data of the user at the user side and uploads the data after adding perturbation. The perturbed data can be used for third-party statistical queries and further recommendation model use because of the satisfaction of $\epsilon -$ LDP. The middle part is the POI recommendation model. At the bottom part, the specific recommendation of the POI check-ins is also random. So even if an attacker has a prediction result, they cannot accurately infer the user's future location.

The following section we will introduce the CGPP-POI model from two aspects, privacy protection mechanism and recommendation model.

5.1. Privacy Protection Mechanism. Figure 1 shows the check-in information of three users; all of them have checked in at different ranges. From these check-ins, we can see the range of activities that each user often does. As can be seen from the activity range, the activity range of blue user and yellow user is highly overlapped. And as we can see from the lower left corner of the range area, the larger the scope, the more check-ins it covers, centered on one of the blue users' check-ins. These collections of check-ins can be considered public information. However, in the user data, it is regarded as the user's private information. For example, when we open Google Map, we can see many places marked, such as schools, hotels, and parks. These are considered public data. However, in the user's check-in records, they are the user's private information. We will use this public information to predict the user's POI. L is the set of these check-ins. Take the blue user u_i in Figure 1 as an example. We expand the blue check-in at the lower left. When $r = r_2$, $\tilde{l}_{u_i,j}$ contains two real check-ins, $r = r_3$, $|\tilde{l}_{u_i,j}| = 3$. When u_i selects r_3, we randomly choose $\widehat{l}_{u_i,j}$ from $\tilde{l}_{u_i,j}$ to replace l_j. The other check-ins for the u_i are replaced as described above then uploaded to the third-party unified collection.

Algorithm 1 satisfies the stochastic perturbation mechanism of LDP, and the derivation process is shown in Definition 6. Except LDP processing before uploading, the specific check-in mechanism is also added to the random in the recommended stage. The purpose of this is to prevent attackers from using the recommendation results to attack users. The pseudo code of related algorithm privacy protection of CGPP-POI is as follows.

5.2. CGPP-POI Recommendation Model. The data of each user is processed by Algorithm 1 and uploaded to the third party. The third party collects the data of all users, analyzes the data information, and constructs the recommendation model. The recommendation model are built based on location scalability characteristics. First of all, the data are extended uniformly. There are several reasons for this: (1) generalize the check-in coordinates to increase data information; (2) with extended check-in, the overlapping information among users increases; and (3) the extended scope may contain the real data of the user, which reduce the interference of disturbed data. In other words, $l_{ui,j}$ and $l_{u\hat{}i,j}$ may be the same extended check-in $l\sim_j$. So, there is more real information in the data, and it is easier to predict u_i's real preferences. Equation (6) is used to obtain the similarity between users. It acts as the weight value between users in the adjacency matrix. Through community detection, users with high similarity are divided into the same community, which can reduce data sparsity. According to the users and the check-in $L\sim$ within the community, the bipartite graph is constructed to obtain the relationship matrix m_n. The recommendation $LA_{ui} = \{l\sim_1, l\sim_2, \cdots, l\sim_j\}$ is the recommendation list for u_i, which includes j extended check-ins $l\sim_j$, and each $l\sim_j$ includes k -specific check-ins. To obtain the recommended list A of extended check-ins by sorting algorithm, we are going to pick a random number of these k check-ins and collect them into the recommendation list B for u_i. The recommendation algorithm of CGPP-POI is described as follows:

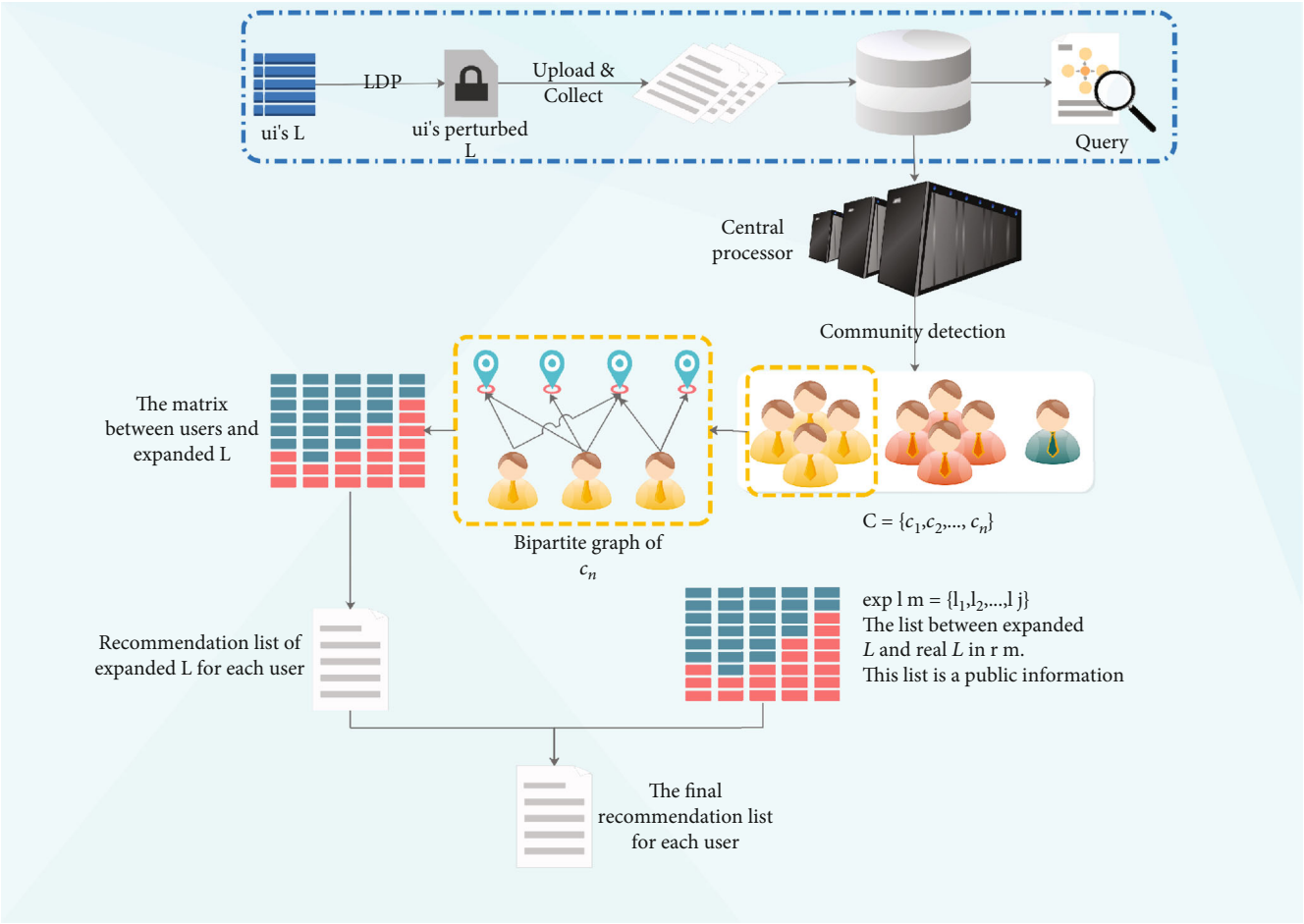


Figure 2: The framework of CGPP-POI.

Input: $l_{u_i} = \{l_{u_i,1}, l_{u_i,2}, \cdots, l_{u_i,j}\}$;
$r = \{r_1, r_2, \cdots, r_m\}$;
$\tilde{L} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}q\}$
Output: $\hat{l}_{u_i} = \{\hat{l}_{u_i,1}, \hat{l}_{u_i,2}, \cdots, \hat{l}_{u_i,j}\}$
1: u_i selects the extension radius r_m
2: **for** $j \in l_{u_i}$ **do**
3: $\tilde{l}_{u_i,j} = l_{u_i,j} \times r_m = \{l_1, l_2, \cdots, l_k\}$
4: $P(\hat{l}_{u_i,j} = l_k) = 1/k$
5: $\hat{l}_{u_i,j}$ replace $l_{u_i,j}$
6: **end for**
7: Get $\hat{l}_{u_i} = \{\hat{l}_{u_i,1}, \hat{l}_{u_i,2}, \cdots, \hat{l}_{u_i,j}\}$

Algorithm 1: Privacy protection of CGPP-POI.

6. Experiments

6.1. Datasets. The datasets we use in this paper are the Gowalla dataset and Brightkite dataset. They are real-world location-based social network, and both of them contain friend relationship information. However, in the preprocessing stage, we only retain the user's check-in information. The check-in information includes the check-in label *LOC*, the corresponding coordinates *LOT*, and the time information. The time information is kept for experimental purposes only and used to divide the train set and test set. The time context is not involved in data publishing and POI recommendation. Table 1 shows the collated information for the two datasets.

The Gowalla dataset is a location-based social network sourced from Stanford University and collected using public APIs. It is an undirected network containing users' location information by check-ins. It consists 196,581 nodes and 950,327 edges. The data were collected from February 2009 to October 2010. A total of 6,442,890 users were collected. The raw data consists two texts, one is the user's friendship and the other is the user's check-in information. The check-in information includes the user ID, check-in time, check-in label, and the corresponding exact coordinates.

The Brightkite dataset is also a location-based social network. Similar to Gowalla, users can check in to share their location. The data is collected based on the site's public API. It contains 58,228 nodes and 214,087 edges. The network was originally a directed graph, but the collector made it an undirected network. In this paper, friend

Input: $G = (\widehat{U}, E)$;
$\widehat{L}_{u_i,j} = \{\widehat{l}_{u_i,1}, \widehat{l}_{u_i,2}, \cdots, \widehat{l}_{u_i,j}\}$;
$\tilde{L} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_m\}$;
$\tilde{l}_q = \{l_1, l_2, \cdots, l_k\}$;
Output: Recommendation list LB_{u_i};
To choose the extend range r_m
$\tilde{L} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_m\}$
for $i \in U$ **do**
    $\tilde{L}_{u_i} = \{\tilde{l}_{u_i,1}, \tilde{l}_{u_i,2}, \cdots, \tilde{l}_{u_i,j}\}$
end for
$Sim(i, j) = \tilde{L}_{u_i} \cap \tilde{L}_{u_j} / \tilde{L}_{u_i} \cup \tilde{L}_{u_j}$
$G' = G \cap Sim$
Initialize each node to a separated community
repeat
    for $i \in V$ **do**
        for $j \in V$ **do**
            Remove i from its community, place to j's community
            Compute the composite modularity gain Δ
        end for
        Choose j with maximum positive gain (if exists) and move i to j's community
        Otherwise, i stays in its community
    end for
until No further improvement in modularity
Get $C = \{c_1, c_2, \cdots, c_n\}$, which are also independent subgraphs
if the number of users in $c_n > 1$ **then**
    $m_{c_n} = U_{c_n} \times \tilde{L}_{c_n}$
    Sort the recommended label for each user, and get recommendation list $LA_{u_i}, \tilde{l}_j \in LA_{u_i}$
else
    Sort by number of check-ins $\tilde{L}_{u_i}$ based on $u_i's$ history. And take the first n and build the recommendation list A.
    end if
for $i \in U$ **do**
    for $j \in LA_{ui}$ **do**
        $P(l_j = l_k) = 1/|\tilde{l}_j|$, to choose n $l_j s$, and LB_{u_i} is obtained.
    end for
end for

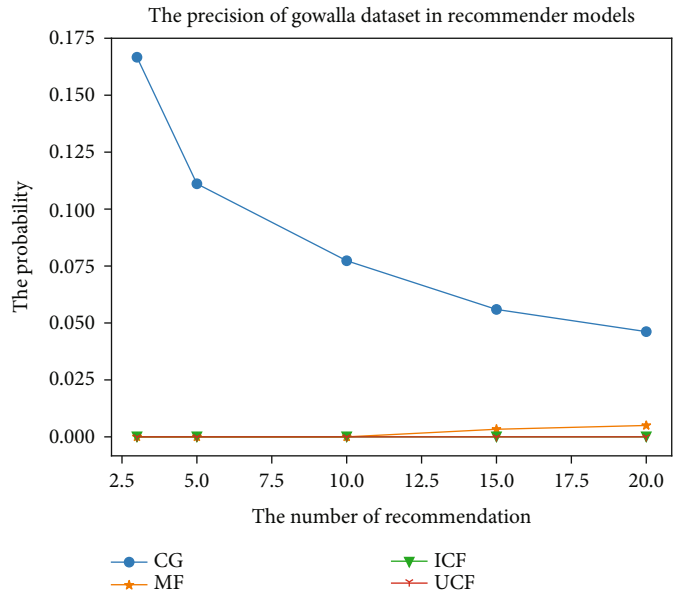
Algorithm 2: The recommendation of CGPP-POI.

Table 1: General statistics about the two datasets.

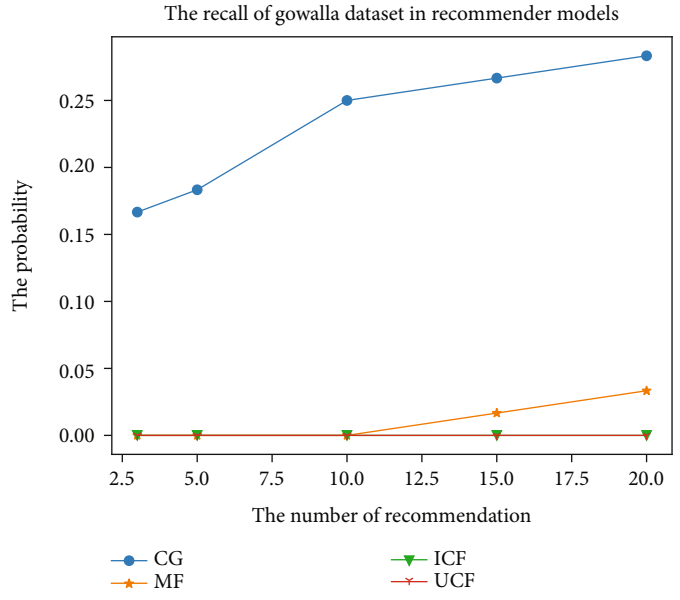
Network property	Gowalla	Brightkite
Number of nodes	196,591	58,228
Number of edges	950,327	214,078
Number of check-ins	6,442,890	4,491,143
Average number of check-ins per user	254	92
Maximum check-in number of user	2175	2100
Minimum check-in number of user	1	1

links are not used and will be deleted during preprocessing. The data was collected from April 2008 to October 2010, and a total of 4,491,143 users were collected. The original data set contains two files, one is a user's check-in file and the other is a friend relationship file. We only keep the first file for experiments. The check-in file contains the user ID, check-in time, specific latitude and longitude information, and check-in label.

6.2. Evaluation Methods. The main research objective of this paper is POI recommendation. Meanwhile, privacy protection is added to the original data to ensure the privacy of user's location. In terms of recommendation, the evaluation criteria for the quality of traditional models are usually taken as the evaluation index from four aspects: accuracy, recall rate, coverage rate, and popularity. POI recommendation, in principle, is roughly the same as item recommendation. The accuracy measures whether the prediction of the user's future behavior at the next moment is accurate. The recall rate judges whether the same check-in is visits multiple times. Popularity can be thought of as areas where users like to go. These measurements have significant guidance for the construction of other project models by third parties. At the same time, for users, people generally like going to places with many people, e.g., business districts and tourist areas. Therefore, adding these recommendations to the list provides a convenient service for users. The specific formulas are as follows:

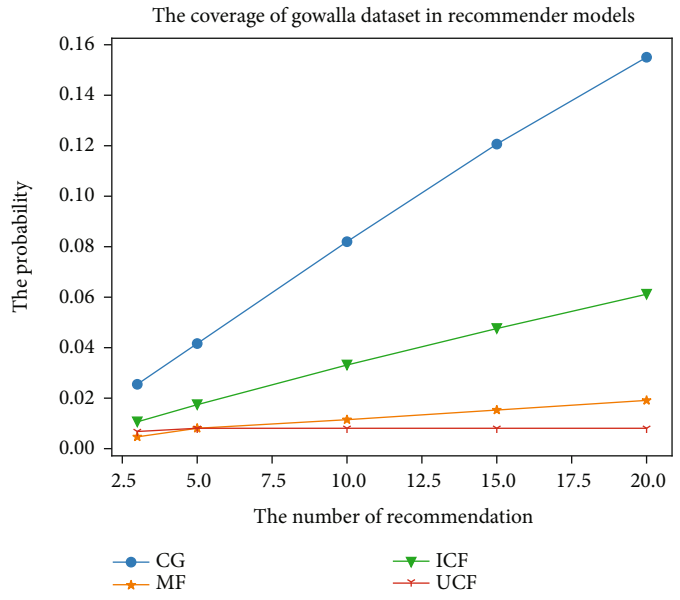


(a)

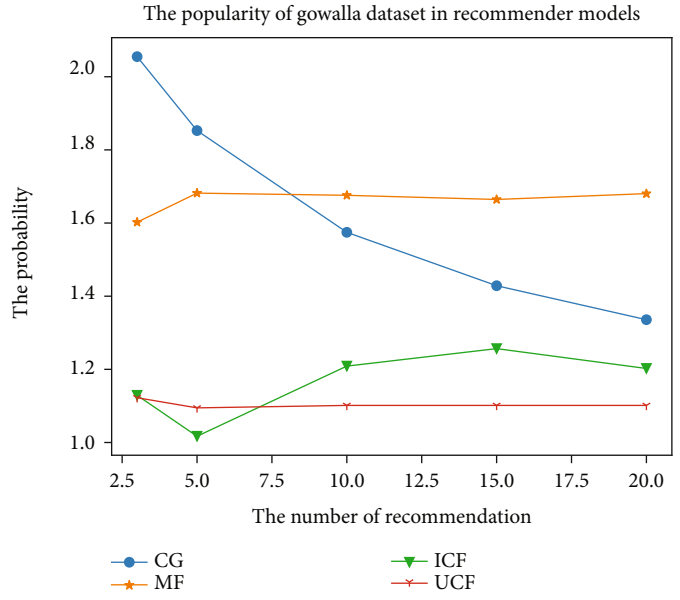


(b)

Figure 3: Continued.

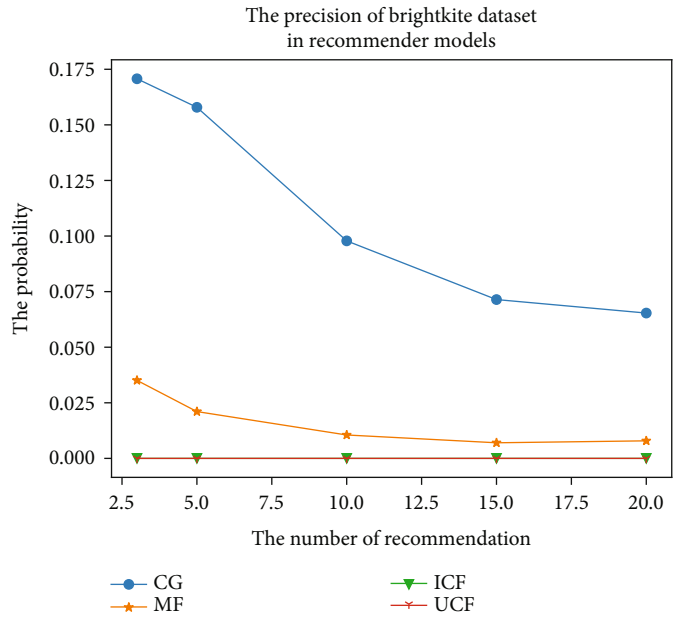


(c)

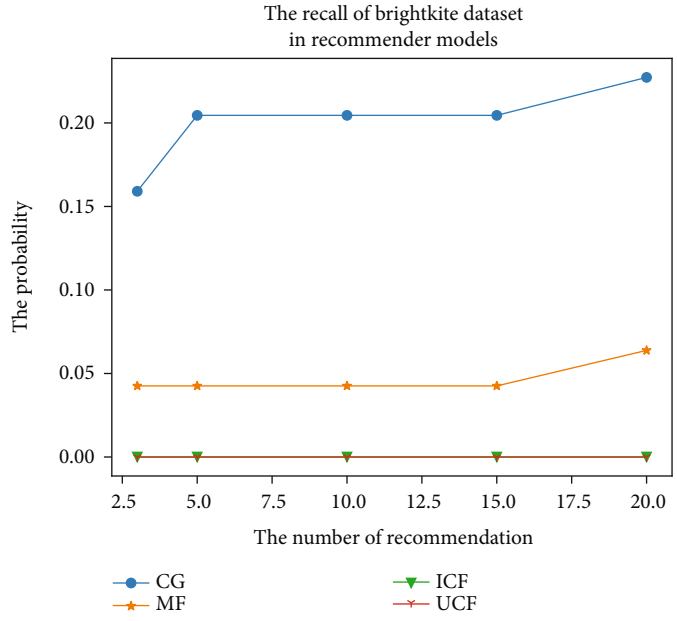


(d)

Figure 3: Continued.

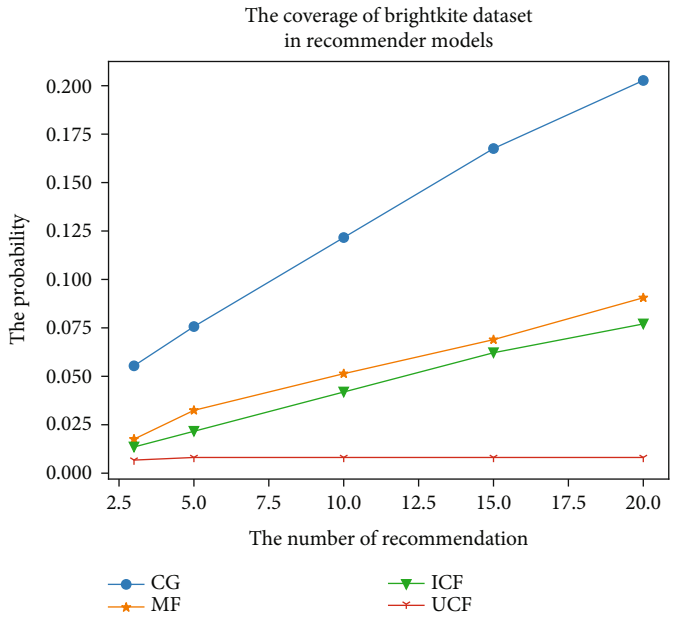


(e)

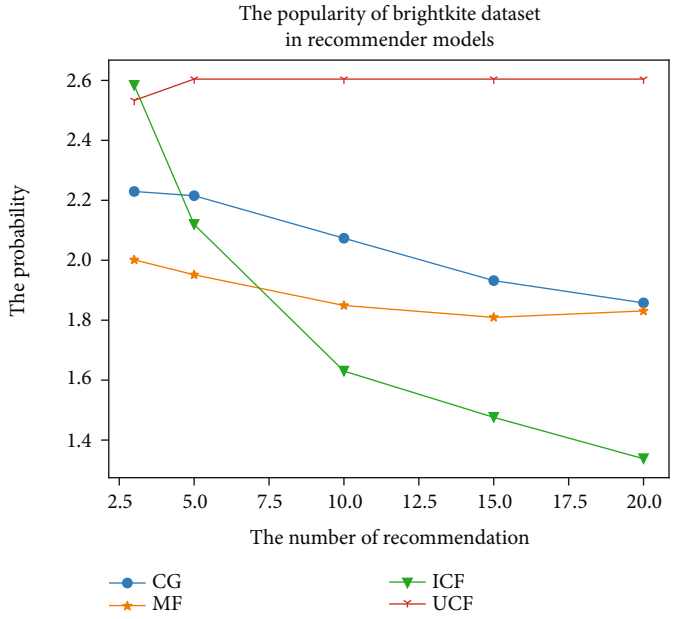


(f)

Figure 3: Continued.



(g)



(h)

Figure 3: The recommendation result of the Gowalla and Brightkite datasets based on real check-ins. (a–d) Results of the Gowalla dataset. (e–h) Results of the Brightkite dataset.

Accuracy is as follows:

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}. \tag{15}$$

Recall rate is as follows:

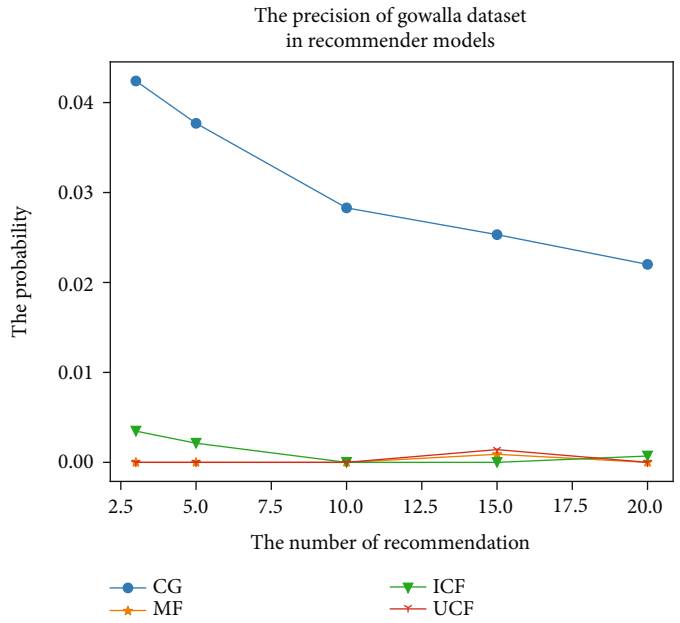
$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}. \tag{16}$$

Coverage rate is as follows:

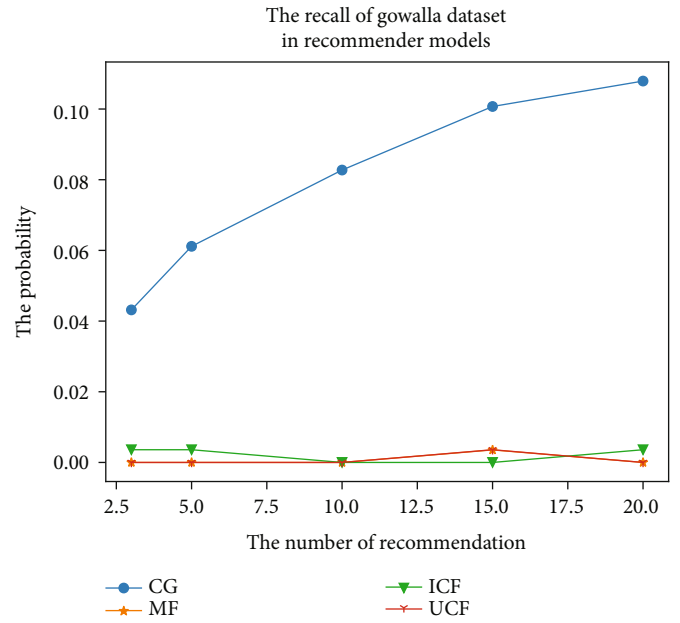
$$\text{Coverage} = \frac{U_{u \in U} R(u)}{LOC}. \tag{17}$$

Popularity is as follows:

$$\text{Popularity} = \frac{1}{U} \sum_{l_j \in R(u)} \frac{d_{l_j}}{|R(u)|}. \tag{18}$$

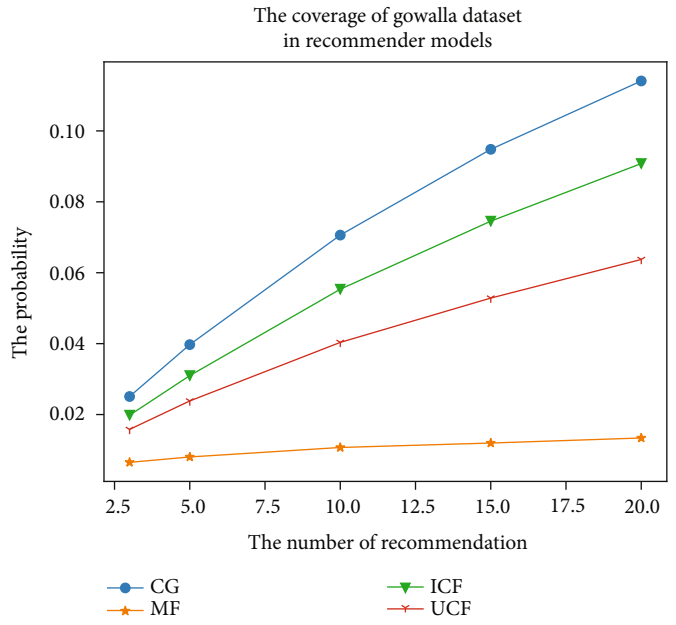


(a)

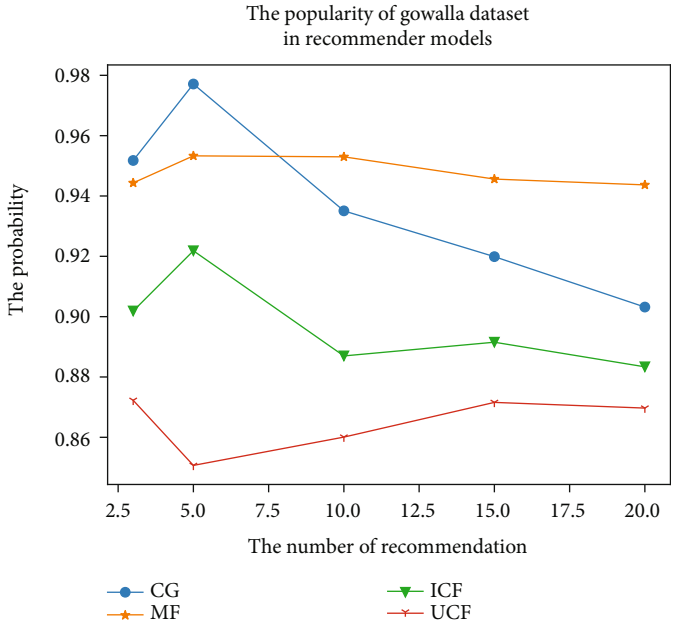


(b)

Figure 4: Continued.

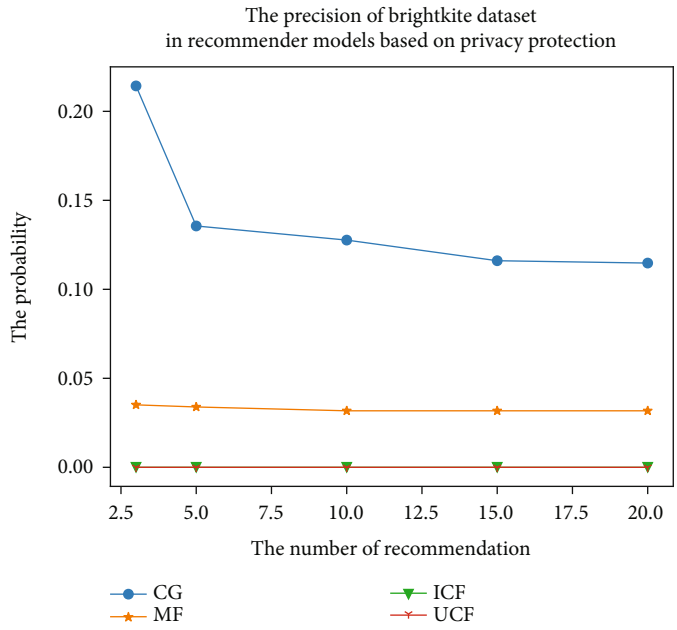


(c)

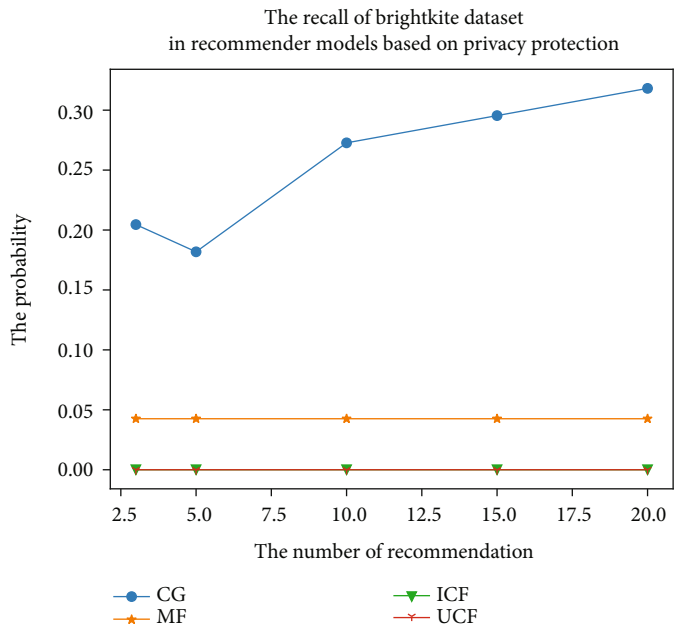


(d)

Figure 4: Continued.

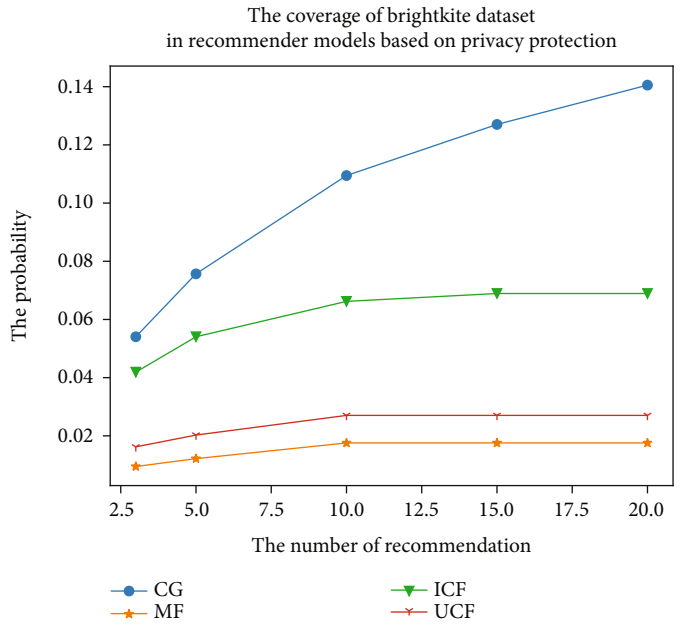


(e)

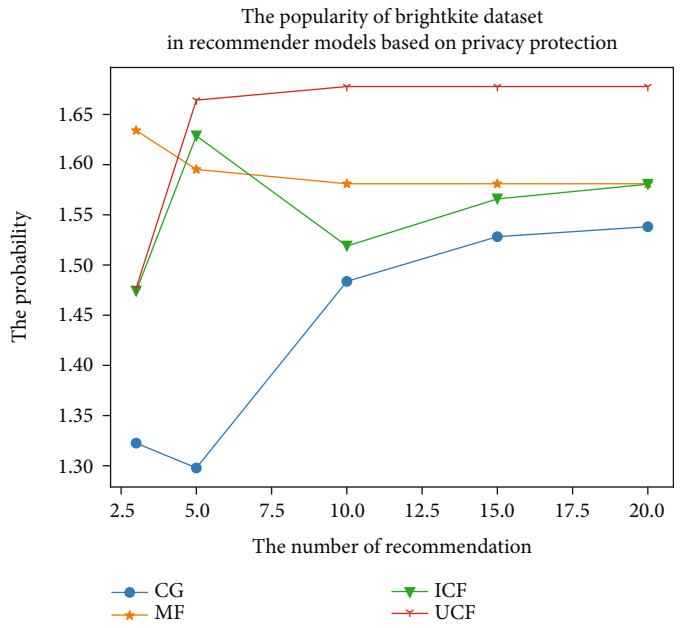


(f)

Figure 4: Continued.



(g)



(h)

Figure 4: The recommendation result of the Gowalla and Brightkite datasets based on privacy protection. (a–d) Results of the Gowalla dataset. (e–h) Results of the Brightkite dataset.

$R(u)$ is a list of check-ins of length n recommended by the recommendation model to each user. $T(u)$ is the actual list of users. U is the collection of users, and L is the set of check-in. d_{lj} is how many people have visited l_j, which represents the popularity of the place.

In addition to evaluating the recommended results, we also evaluate the impact of different extension ranges on recommendations.

6.3. Experimental Setup. In this section, the specific experimental settings and some operations in the experiment are addressed. In the preprocessing stage, we deleted the friend relationship and time information in the original data. The training set and test set are divided according to the number of users, and the partition ratio is 7 : 3. Then, we retain the most recent three check-in records of users to compare with the predicted results. We choose three benchmark algorithms: item-based collaborative filtering, user-based collaborative filtering, and MF as the baseline algorithms. The reasons for choosing these three are as follows. (1) These three algorithms are recommended classical algorithms; (2) the location-based recommendation in this paper is not path

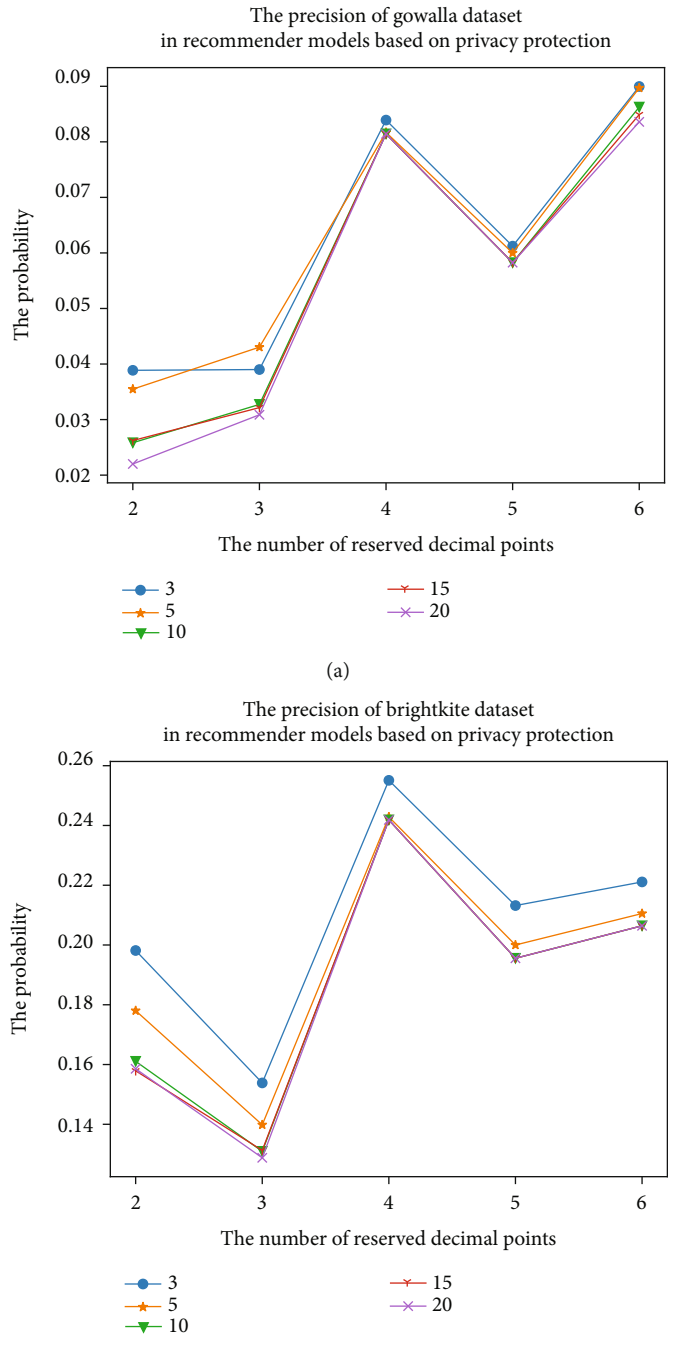


Figure 5: Privacy protection results in different degrees. (a) Result of the Gowalla dataset. (b) Result of the Brightkite dataset.

prediction, so many algorithms are not applicable; and (3) these three comparison algorithms have good results in various indicators, and the prediction results are stable and widely applicable in the recommendation algorithm. The number of neighbors in CF is set to 3. The recommended number of LA_{ui} is 3. The recommended number of LB_{ui} is set to $\{3, 5, 10, 15, 20\}$. r_m is the number of decimal points retained in the experiment. We round r_m to the nearest hundredth for latitude and longitude in recommendation experiment. While in the experiment on the impact of the extended range on the recommended results, the number of decimal points retained for latitude and longitude is $r_m = \{2, 3, 4, 5, 6\}$.

6.4. Experimental Results. We perform two types of experiments. The first one is the recommendation experiment. The experiment sets up two specific scenarios in which the evaluation results were obtained. One scenario is preprocessing recommendations without privacy protection. Another scenario is a recommendation experiment to perturb the data during processing. The second experiment is to explore the impact of the intensity of privacy protection on

recommendations. The results are given in the following two sections.

6.4.1. Recommendation Results. Figures 3 and 4 both contain eight subfigures. Figures 3(a)–3(d) and 4(a)–4(d) are the results of the Gowalla dataset. Figures 3(e)–3(h) and 4(e)–4(h) are the results of the Brightkite dataset. We evaluated our proposed method and three baselines from four aspects of accuracy, recall rate, coverage rate, and popularity. Figure 3 is the recommendation result without noise, and Figure 4 is the recommendation result with noise. We can see from these figures that, except for popularity, the results of CGPP-POI algorithm are all superior to the other three evaluation indexes. The reason that CGPP-POI's result is less popular than other algorithms is that CGPP-POI divides users into communities, where recommendation results are obtained from. Some communities have a low level of popularity, which can lead to a decrease in popularity. Other algorithms do not have the concept of community, so it is a global selection of check-in recommendation and can derive a higher popularity. We can see the accuracy from Figures 3 and 4. The accuracy after adding noise is affected and is much lower than without noise. That is because we choose to round to the nearest hundredth. That is, the larger extended range and more perturbation will lead to the lower accuracy. Accuracy also decreases as the number of recommendations increases. This is because we are setting the number of comparison check-ins to be reserved at 3. So, as the number of recommendations increases, the accuracy decreases. As can be seen from the recall rate, users are more likely to visit places they have been before.

6.4.2. The Impact of r_m on Recommendation. Figure 5 shows the changes of recommendation accuracy under different privacy protection levels. Figure 5(a) shows the predicted results of the Gowalla dataset, and Figure 5(b) shows the predicted results of the Brightkite dataset. $r_m = \{2,3,4,5,6\}$ is the number of digits reserved after the decimal point. The smaller the number, the wider it extends in the coordinates. The more information it contains, the more perturbations it has. As can be seen from the results of the two figures in Figure 5, the bigger extension range results in a lower the accuracy. The prediction accuracy of $r_m = 3$ is lower than when $r_m = 2$, because the community division does not work well at $r_m = 3$. It can be seen from the two figures that when the number of recommendation is 3, the accuracy is higher than other numbers. This illustrates that (1) both the number of recommendation and the reserved decimal points will impact the result and (2) the higher the accuracy, the closer the recommended quantity is to the reserved comparison quantity. Therefore, it is not necessary to recommend a lot of information to the user, which will increase the time cost of the user.

7. Conclusions

In this paper, we proposed CGPP-POI which is a recommendation model with privacy protection mechanism. Through this model, users can set their own privacy protection scope and then upload data to the third party after LDP. The location information of the user can be protected through LDP processing, while the data remains acceptable utility. The data uploaded after perturbation can satisfy the LDP in the statistical query and maintain its statistical results. But when we try to use this data to build recommendation models, the results are terrible because every user's preferences is different. Therefore, we generalize the disturbed data and build the community by taking advantage of the overlap between users after location expansion. We enhance the coupling between users with high similarity and search for the relationship between users and check-in within the community. Finally, the obtained recommendation results satisfy local differential privacy. We compare the baseline recommendation algorithms on two real data and found that it was superior to the baseline recommendation algorithm in terms of accuracy, recall rate, coverage, and popularity. We also explore the impact of different privacy intensities on recommendations. As can be seen from the experimental results, the bigger the extension range, the lower the accuracy. Our future goal is to improve the recommendation model of POI based on privacy protection by extending model parameters. We aim to improve the accuracy of prediction and protect user location information simultaneously.

Data Availability

The data used to support the research is generally unavailable due to public releasability constraints. Please contact the corresponding author for special release consideration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Our research is funded by the Fundamental Research Funds for the Central Universities (No. 3072021CF0609).

References

[1] Q. Hou, M. Han, and Z. Cai, "Survey on data analysis in social media: a practical application aspect," *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 27–47, 2020.

[2] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: enhanced privacy protection in location based services," in *International Symposium on Location-and Context-Awareness*, pp. 70–87, Berlin, Heidelberg, 2009.

[3] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[4] https://en.wikipedia.org/wiki/aol_search_data_leak.

[5] https://www.wired.com/2009/12/netflix-privacy-lawsuit/.

[6] Q. Luo, X. Cheng, J. Yu, Z. Cai, D. Yu, and L. Zhang, "Fast skyline community search in multi-valued networks," *Big Data Mining and Analytics*, vol. 3, no. 3, p. 171, 2020.

[7] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *Acm Sigkdd Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[8] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of pointof-interest recommendation in location-based social networks," *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.

[9] P. Zhao, A. Luo, Y. Liu et al., "Where to go next: a spatio-temporal gated network for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, pp. 5877–5884, 2019.

[10] G. Li, G. Yin, J. Yang, and F. Chen, "Sdrm-ldp: a recommendation model based on local differential privacy," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6640667, 15 pages, 2021.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[12] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, Las Vegas, Nevada, USA, 2008.

[13] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, pp. 1257–1264, 2007.

[14] C. Huang, Y. Cheng, W. Zhou, and J. Jia, "Web3d learning framework for 3d shape retrieval based on hybrid convolutional neural networks," *Tsinghua Science and Technology*, vol. 25, no. 1, p. 93, 2020.

[15] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, "Adapting to user interest drift for poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2566–2581, 2016.

[16] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, "A trust-based probabilistic recommendation model for social networks," *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.

[17] G. Li, Z. Cai, G. Yin, Z. He, and M. Siddula, "Differentially private recommendation system based on community detection in social network applications," *Security and Communication Networks*, vol. 2018, Article ID 3530123, 18 pages, 2018.

[18] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *21st International Conference on Data Engineering Workshops (ICDEW'05)*, pp. 1248–1248, Tokyo, Japan, 2005.

[19] P. Shankar, V. Ganapathy, and L. Iftode, "Privately querying location-based services with sybilquery," in *Proceedings of the 11th international conference on Ubiquitous computing*, pp. 31–40, Orlando, Florida, USA, 2009.

[20] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 237–246, Beijing, China, 2008.

[21] J. Reagle and L. F. Cranor, "The platform for privacy preferences," *Communications of the ACM*, vol. 42, no. 2, pp. 48–55, 1999.

[22] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest neighbor search with strong location privacy," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 619–629, 2010.

[23] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 121–132, Vancouver, Canada, 2008.

[24] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.

[25] I. Memon, Q. A. Arain, M. H. Memon, F. A. Mangi, and R. Akhtar, "Search me if you can: multiple mix zones with location privacy protection for mapping services," *International Journal of Communication Systems*, vol. 30, no. 16, article e3312, 2017.

[26] T. Hashem and L. Kulik, ""Don't trust anyone": privacy protection for location-based services," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 44–59, 2011.

[27] Z. Chen, X. Hu, J. Xiaoen, and K. G. Shin, "LISA: location information scrambler for privacy protection on smartphones," in *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 296–304, National Harbor, MD, USA, 2013.

[28] Z. Cai, Z. Duan, and W. Li, "Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 8, pp. 2576–2591, 2021.

[29] Z. Cai and Z. He, "Trading private range counting over big iot data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[30] Z. Xu and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial iots," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[31] Z. Cai and Z. Xu, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[32] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[33] C. Dwork, "Differential privacy: a survey of results," in *International Conference on Theory and Applications of Models of Computation: Theory and Applications of Models of Computation*, Springer, 2008.

[34] X. Zhao, Y. Li, Y. Yuan, X. Bi, and G. Wang, "Ldpart: effective location-record data publication via local differential privacy," *IEEE Access*, vol. 7, pp. 31435–31445, 2019.

[35] Y. S. R. Xin, J. Zhang, and Y. Shao, "Complex network classification with convolutional neural network," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447–457, 2020.

[36] E. J. Newman Mark, "Networks: an introduction," *Astronomische Nachrichten*, vol. 327, no. 8, pp. 741–743, 2010.

[37] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, article 066133, 2004.

[38] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, Springer, 2006.

[39] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103, Providence, RI, USA, 2007.